PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 15/00, 17/00, 17/21, 17/24, 7/00 — A1

(11) International Publication Number: **WO 00/63783**

(43) International Publication Date: 26 October 2000 (26.10.00)

(21) International Application Number: PCT/US00/07792

(22) International Filing Date: 24 March 2000 (24.03.00)

(30) Priority Data:
09/294,701 19 April 1999 (19.04.99) US

(71) Applicant: LIAISON TECHNOLOGY, INC. [US/US]; Suite D400, 11044 Research Boulevard, Austin, TX 78759 (US).

(72) Inventors: MIRANKER, Daniel, P.; 5225 Fossil Rim Road, Austin, TX 78746 (US). OBERMEYER, L., Lance; 5117 Jekins Cove, Austin, TX 78730 (US). NAVRATIL, Paul, A.; 2600 Rio Grande Street, Austin, TX 78705 (US).

(74) Agent: HULSEY, William, N., III; Gray Cary Ware & Freidenrich, Suite 1440, 100 Congress Avenue, Austin, TX 78701 (US).

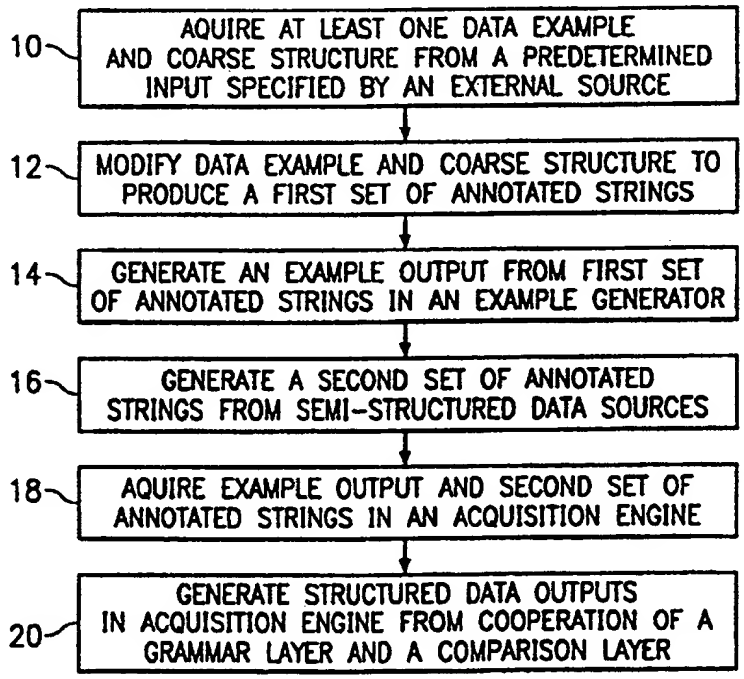
(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD AND SYSTEM FOR GENERATING STRUCTURED DATA FROM SEMI-STRUCTURED DATA SOURCES

(57) Abstract

A system and method for generating structured data outputs from a semi-structured data source. The steps of this method include generating an example output from an example generator (14). The example output is generated in response to the acquisition of a sequence of annotated strings (12). The annotated strings are generated in response to the acquisition and modification of at least one data example and corresponding coarse structure from a predetermined input source (10). Also, a second sequence of annotated strings is generated from input from a semi-structured data source (16). Both the example output and the second sequence of annotated strings are input to an acquisition engine (18) that implements a grammar layer incorporating a top-down parsing method and a comparison layer. The structured data outputs are generated through the cooperation of the comparison layer and the grammar layer (20).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

METHOD AND SYSTEM FOR GENERATING STRUCTURED DATA FROM SEMI-STRUCTURED DATA SOURCES

TECHNICAL FIELD OF THE INVENTION

The present invention relates generally to data acquisition and structuring systems and methods, and more particularly, a system and method for generating structured data outputs from semi-structured data inputs.

BACKGROUND OF THE INVENTION

The general field of this invention relates to generating structured data outputs from semi-structured data inputs. A particular application of the invention is acquiring and structuring data to form virtual internet databases. Virtual internet databases are databases whose content is owned, stored and managed on servers distributed across a computer network.

Recently, internet usage and access has increased markedly. The availability and quantity of information on the internet has also increased. Many software products that can produce printed reports can now produce WEB reports. These products produce reports that may be displayed on a WEB page. This is accomplished by embedding the text of the report within the computer language called HTML. Although posted reports and information appear as data on the WEB page, this HTML representation is not a data representation. Rather, the WEB browser serves as a vehicle to display information much like that of a page in a textbook. This presents the problem of incompatibility between the HTML representation and the PC desktop and server applications. Ultimately, the current practice of

employing WEB browsers has reduced PCs back to "dumb" terminals. The graphics may be exciting, but functionally all the computing power is limited to providing users with little more than a sophisticated data viewing window.

Several methods have been developed to address the problem of moving semi-structured data from the internet to a PC or server application. These methods include *ad hoc* engineering methods, Graphical User Interface (GUI) methods, and machine learning methods.

Ad hoc methods entail writing specialized parsing programs in a language such as PERL or LEX to extract the necessary information. These types of programs are called wrappers. A wrapper is a software method that converts data such as HTML code into structured data for further processing. These types of programs employ the use of regular expressions in the parsing process. Unfortunately, these *ad hoc* methods are labor intensive. Depending on the skill of the programmer and the complexity of the particular job, these methods can take days to develop. Also, these methods are not an option for an average internet user with no formal training or knowledge of HTML and programming methods.

Due to the tedious nature of custom wrapper design, further methods have been developed that employ GUIs to facilitate the wrapper generation. The GUI hides all the engineering details beyond the extracted data pattern definitions. Like the *ad hoc* methods discussed above, these packages implement regular expression parsing algorithms. In general these methods require some knowledge of both HTML and regular expressions, therefore they may not be suitable to some internet users.

Due to the use of regular expressions, both *ad hoc* methods and GUI methods can result in what is called brittle parses. Brittle parses result when changes in format of the HTML page cause the parse to fail. A single format change is not guaranteed to break the parse, but the likelihood is sufficiently high as to prevent any guarantees of robust behavior.

Recently, machine learning methods have been developed to address the need for engineering skills in the development of wrappers. Given a set of similar WEB pages and an example of the data to be parsed from each page, these methods automatically generate a wrapper. Unfortunately, these methods require a large number of examples to reliably produce wrappers. An example of such a method can be found in *A Hierarchical Approach to Wrapper Induction*, Muslea, et al. (1999). This method may require 8-10 examples to produce the wrappers. The generated wrappers are based on regular expression techniques and are brittle. Although these wrappers may work for format changes known prior to wrapper generation, they may fail on empirical format changes as the regular expression based methods discussed above.

Ideally, it is desirable to develop a method for a user to gain access to semi-structured data for a PC or server application without requiring the user to have previous knowledge HTML or regular expressions. In addition, it is advantageous if the method does not require the enumeration of examples covering possible format changes.

SUMMARY OF THE INVENTION

The present invention provides a system and method for acquiring and structuring data from semi-structured data sources that substantially eliminates or reduces disadvantages and problems associated with previously developed systems and methods used for developing structured data sources from on-line sources such as the Internet, intranets, or other network systems.

More specifically, the present invention provides a system for generating structured data outputs from semi-structured data sources. The steps of this method include generating an example output from an example generator. The example output is generated in response to the acquisition of a sequence of annotated strings. The annotated strings are generated in response to the acquisition and modification of as little as one data example and a corresponding coarse structure from a predetermined input source. Also, a second sequence of annotated strings in generated from input from a semi-structured data source. Both the example output and second sequence of annotated strings are input to an acquisition engine that implements a grammar layer incorporating a top-down parsing method and a comparison layer. The structured data outputs are generated through the cooperation of the comparison layer and the grammar layer.

The present invention provides an important technical advantage in that it does not require the user to have knowledge of HTML or knowledge of pattern matching languages. The graphical interface guides the user through a set-up phase and completely hides all technical details.

The present invention provides an important technical advantage in that it requires only one single data example. Once this set-up process is complete, the acquisition engine can be pointed to related WEB pages, as well as updated versions of the same page, and it will automatically extract data and route it to applications.

The present invention provides yet another technical advantage in that the system is able to cope with the format changes from the source pages, including changes in the order of data values. Thus, the technology produces reliable results even when the data sources are reformatted, updated or amended by the content providers.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numerals indicate like features and wherein:

FIGURE 1 is a flow diagram of one embodiment of the present invention;

FIGURE 2 is a block diagram of the gross architectural breakdown 10 of an embodiment of the present invention;

FIGURE 3 is a flow diagram for the generation of HTML phonemes of the embodiment of FIGURE 1;

FIGURE 4 illustrates the decomposition of HTML stings into tokens and phonemes;

FIGURE 5 is an example of the GUI used to extract example data items and the corresponding structure; and

FIGURE 6 represents an example of the pattern dictionary including patterns of phonemes and the corresponding terminals of the context free grammar.

DETAILED DESCRIPTION OF THE INVENTION

Preferred embodiments of the present invention are illustrated in the FIGUREs, like numerals being used to refer to like and corresponding parts of the various drawings.

The present invention provides a system and method for generating structural data outputs from semi-structured data inputs. Details of embodiments of the present invention are discussed below.

FIGURE 1 is the flow diagram of one embodiment of the present invention. At step 10, at least one data example and coarse structure is acquired from a predetermined input specified by an external output. The at least one data example can be exactly one data example. This predetermined input serves to present an example of the type of data to be acquired and structured. Such a type of predetermined input can be PDF files, semi-structured text files, or HTML files. The external source can be a storage means such as a database server or a WEB server. At step 12, the data example and coarse structure are modified to produce a first set of annotated strings. These annotated strings serve as data structures providing one or more attributes regarding each data example and the coarse structure.

At step 14, an example output is generated in an example generator from the first set of annotated strings. The example output comprises a pattern dictionary

containing at least one annotated string associated with a terminal that represents a terminal of a context free grammar. At step 16, a second set of annotated stings are generated from at least one semi-structured data source. The semi-structured data sources serve as the source of data which is to be acquired and structured. A source for such semi-structured data sources can be a database server or WEB server. At step 18, the example output and the second set of annotated strings are acquired by the acquisition engine. In turn, at step 20, the acquisition engine generates the structured data outputs from cooperation of a grammar layer and a comparison layer contained within the acquisition engine. The grammar layer and comparison layer work in cooperation to locate in the second set of annotated strings the desired data outputs based on the example output from the example generator.

FIGURE 2 is a block diagram of the gross architectural breakdown 20 of one embodiment of the present invention. The gross architectural breakdown 20 can be divided into two major parts: the training stage 26 and the acquisition stage 28. The internet 24 provides both the training stage 26 and the acquisition stage 28 with an input WEB training page 32 to be used to extract an example and train the system as to the type of information and format of information desired. The internet 24 also provides the acquisition stage 28 with the semi-structured data sources, incoming HTML pages 34, to search for and structure the type of data specified by the training stage 26. These two stages, the training stage 26 and the acquisition stage 28, can be further broken down. The training stage 26 is comprised of a GUI 36, preprocessor 38, and a builder 46

containing an example generator 48. The GUI 36 is used to extract information from the input WEB training page 32 located on the internet 24. The preprocessor 38 then interfaces with the GUI 36 to produces HTML phonemes 40 representing the extracted information from the input WEB training page 32.

The HTML phonemes 40 are input to the example generator 48. Example generator 48 converts the HTML phonemes 40 into a series of patterns which populate a pattern dictionary 50 and generates a context-free grammar 52. Patterns in the pattern dictionary 50 may include the user input with phonemes 40 on each side and a corresponding weight for each phoneme. There can be multiple patterns in the pattern dictionary 50. The pattern dictionary 50 and the context-free grammar 52 are then input into the acquisition stage 28, specifically the acquisition engine 54. HTML phonemes 44 generated from the incoming HTML page 34 through the use of a preprocessor 42, are also input into the acquisition engine 54. The acquisition engine 54 can be broken down further into a grammar layer 56 and a comparison layer 58. The pattern dictionary 50 and a context free grammar 52 are used to extract the structured data outputs 30 contained within the HTML phonemes 44. These structured data outputs 30 are outputs of the acquisition engine 54.

FIGURE 3 is a flow diagram for the generation of HTML phonemes 40, 44. A pure HTML representation 62 of the incoming HTML information from the GUI 36 or the incoming HTML page 34 is created from step 60. The incoming HTML information may contain scripts and/or call backs to the web-server, so called active components. At step 60, these

active components of the incoming HTML information are converted to a pure HTML representation 62 of the HTML information. In turn, lexical analysis is performed at step 64 by breaking the pure HTML representation 62 into substrings called tokens 66. The tokens 66 are then adorned with characteristic features at step 68 which outputs the HTML phonemes 40, 44. These characteristic features include, but are not limited to, markups that change font size, markups that add hyperlinks, strings types, row and column number of HTML table cells associated with strings, and row and column numbers of table cells with respect to the presentation of the semi-structured data within the incoming HTML information from the GUI 36 or the incoming HTML page 34.

FIGURE 4 illustrates an example of decomposing an incoming HTML string 70 from the incoming HTML page 34 into a token list 72. The HTML phonemes chart 74 depicts each token 66 in the income HTML string 70 with its corresponding characteristic features. Each token 66 and its characteristic feature is called an HTML phoneme 44.

FIGURE 5 is a representation of the GUI 36 used to extract information for the generation of the context-free grammar 52 in the pattern dictionary 50. The GUI 36 provides the example generator 48 with a coarse structure of the structured data outputs 30 to be acquired. There are multiple coarse structures that will determine the acquisition of the structured data outputs 30. These coarse structures include: one data record, multiple data records from a row major form not necessarily an HTML table, multiple data records from a column major form not necessarily in an HTML table, and nested combinations of

the above three structures, including object-like structures. The GUI 36 provides the example generator 48 with HTML phoneme representations of each example data value and phonemes to distinguish the coarse structure. Overall, the net input to the example generator 48 is a mapping of text in the input WEB training page 32 to data values and a structured record.

FIGURE 6 is an example of the pattern dictionary 34 generated from the example generator 48. Each pattern P_j consists of a sequence of HTML phonemes 40, p_0, $p_1 \ldots p_n$ and a set of corresponding weights w_0, $w_1 \ldots w_n$. A terminal T_j for the context free grammar is assigned in one-to-one correspondence with each pattern in the pattern dictionary. The context free grammar represents the coarse structure and number of data values to be extracted from the semi-structured data source 34. Once the context-free grammar 52 and the pattern dictionary 50 have been generated in the training stage 26, they are passed to the acquisition engine 54. An example of such an engine can be found in *Modification of Earley's Algorithm for Speech Recognition*, NATO ASI Series, Vol. F46, Paeseler, Annedore (1988), which is incorporated by reference herein in its entirety.

The comparison of patterns from the pattern dictionary 50 with an input stream of HTML phonemes 44 from the incoming HTML page 34 occurs in the comparison layer 58. In the comparison layer 58 a matching score between the pattern in the pattern dictionary 50 and a pattern found in the input stream is calculated. This matching score can be calculated using an weighted edit distance algorithm incorporating top-down methods with pruning or dynamic programming. Examples of such weighted edit distance

algorithms can be found in *Pairwise Sequence Alignment*, Geigerich, Robert, and Wheeler, David (last modified May, 1996), <http//:www.techfak.uni-bielefeld.de/bcd/curric/PrwAli/prwali.html>, which is incorporated by reference herein in its entirety. This algorithm incorporates a normalized weighted sum of scores between phonemes from the pattern in the pattern dictionary 50 and a phoneme in the input steam of HTML phonemes 44. Recall patterns in the pattern dictionary 50 may have different phonemes and each phoneme has a corresponding weight. Once the matching score is generated, the matching score and the matching pattern from the input HTML stream is supplied to the grammar layer 56. The grammar layer 56 implements a top-down parsing method based on a set of grammar rules from the context free grammar 52 to determine new patterns which can follow the previously found matching pattern. These new patterns are supplied to the comparison layer 58 to complete patterns at the grammar level from the pattern dictionary 50 with which the input stream of the HTML phonemes is to be compared. The process alternates between the grammar layer 56 and the comparison layer 58 until the last of the HTML phonemes 44 from the incoming HTML page 34 are compared. The structured data outputs 30 are output based on the sequence of patterns that has the best cumulative matching score and corresponds to a correct parse of the document defined by the context free grammar 52.

The present invention has many advantages. First the use of a GUI 36 to extract the training information from the input WEB training page 22 hides all the technical details behind the builder 46 and the acquisition engine

54. These enables the use of the present invention by users with little or no previous knowledge of HTML and parsing methods.

In addition the present invention requires a minimum of one data example from the input WEB training page 32 in the training stage 26 to acquire the desired structured data outputs 30. This eliminates time-consuming processes of presenting multiple examples in order to acquire and structure the desired data outputs 30.

An important advantage is that the present invention is able to cope with format changes from the semi-structured data sources. Changes such as font size, font color, or permutations in the data value will not cause the acquisition engine to fail. The characteristic features which adorn the tokens 66 to create the phonemes 40, 44 reflect properties including but not limited to format. Even if the page has undergone formatting changes, the original data value will still have some best match. Due to the cumulative characteristics of a pattern, the weighted edit distance almost always finds the correct match.

It is important to note that regular grammars are a subset of context-free grammars. Therefore, the present invention will work properly for regular grammars, as well.

In summary, the present invention provides a Method and System for Generating Structured Data from Semi-structured Data Sources. The steps of this method include generating an example output from an example generator. The example output is generated in response to the acquisition of a sequence of annotated strings. The annotated strings are generated in response to the acquisition and

modification of at least one data example and corresponding coarse structure from a predetermined input source. Also, a second sequence of annotated strings in generated from input from a semi-structured data source. Both the example output and second sequence of annotated strings are input to an acquisition engine that implements a grammar layer incorporating a top-down parsing method and a comparison layer. The structured data outputs are generated through the cooperation of the comparison layer and the grammar layer. The present invention is robust to formatting changes and permutations in the semi-structured data sources. In addition, the present invention is easy to use, requiring no prior knowledge of parsing languages or HTML.

Although the present invention has been described in detail, it should be understood that various changes, substitutions and alterations can be made hereto without departing from the spirit and scope of the invention as described by the appended claims.

WHAT IS CLAIMED IS:

1. A method for generating structured data outputs from semi-structured data sources, said method comprising:

generating an example output from an example generator in response to an acquisition of a first plurality of annotated strings, said first plurality of annotated strings generated from an acquisition and modification of at least one data example and a corresponding coarse structure from a predetermined input specified by an external source;

generating a second plurality of annotated strings relating to an input from said semi-structured data sources;

acquiring said example output, and said second plurality of annotated strings in an acquisition engine, said acquisition engine comprising a grammar layer and a comparison layer; and

generating structured data outputs from a cooperation of said grammar layer and said comparison layer, said grammar layer comprising a top-down parsing algorithm.

2. The method of Claim 1, wherein said at least one data example is one data example.

3. The method of Claim 1, wherein said example output is a context-free grammar and a pattern dictionary.

4. The method of Claim 3, wherein said context-free grammar further comprises regular grammar.

5. The method of Claim 1, wherein said acquisition and modification of said at least one data example and said corresponding said coarse structure further comprises separating said at least one data example and corresponding coarse structure utilizing lexical analysis to form a first set of tokens and annotating said first set of tokens with characteristic features to produce said first plurality annotated strings.

6. The method of Claim 5, wherein said predetermined input is a first HTML page and said first set of tokens is a first set of HTML phonemes.

7. The method of Claim 1, wherein the said predetermined input is a PDF file or a semi-structured text file.

8. The method of Claim 1, wherein the step of generating said second plurality of annotated strings further comprises preprocessing said input from said semi-structured data sources to form said second plurality of annotated strings.

9. The method of Claim 8, wherein the step of preprocessing further comprises separating said input from said semi-structured data sources utilizing lexical analysis to form a second set of tokens and annotating said second set of tokens with characteristic features to produce said second plurality of annotated strings.

10. The method of Claim 9, said semi-structured data sources are one or more related HTML pages and said second set of tokens is a second set of HTML phonemes.

11. The method of Claim 1, wherein said acquisition of said at least one data example and said coarse structure from a predetermined input further comprises a user interface for identification of said at least one data example and said coarse structure.

12. The method of Claim 11, wherein said user interface is a Graphical User Interface (GUI) and said predetermined input is an HTML page.

13. The method of Claim 3, wherein the step of generating said pattern dictionary further comprises assigning a pattern consisting of a portion of said sequences of annotated strings to each of said at least one data example and assigning additional patterns to distinguish said corresponding coarse structure from said predetermined input.

14. The method of Claim 13, wherein the step of generating said context-free grammar further comprises generating terminals that are in one-to-one correspondence with said patterns in said pattern dictionary.

15. The method of Claim 1, wherein said cooperation of said grammar layer and said comparison layer further comprising:

sequentially comparing in said comparison layer said patterns in said pattern dictionary against the said second sequence of annotated strings to find a matching pattern in a portion of said second sequence of annotated strings;

compiling a matching score representing a quality of a match between said patterns in said pattern dictionary and said matching pattern;

passing said matching score and said matching pattern to said grammar layer.

extending already found matching patterns with said matching pattern to form a sequence of matching patterns; and

executing a set of grammar rules defined by said context-free grammar on said sequence of matching patterns to locate a legal sequence of strings defined by said set of grammar rules and representing said structured data outputs.

16. The method of Claim 15 wherein compiling the matching score further comprises implementing a weighted edit distance algorithm to calculate the matching score.

17. The method of Claim 16, wherein the weighted edit distance algorithm is a top down method with pruning.

18. The method of Claim 16, wherein the weighted edit distance algorithm is a dynamic programming method.

19. A system comprising a computer program stored in a computer readable form on a tangible storage medium for

generating structured data outputs from semi-structured data sources, the computer program executable to:

generate an example output from an example generator in response to an acquisition of a first plurality of annotated strings, said first plurality of annotated strings generated from an acquisition and modification of at least one data example and a corresponding coarse structure from a predetermined input specified by an external source;

generate second plurality of annotated strings relating to an input from said semi-structured data sources;

acquire said example output, and said second plurality of annotated strings in an acquisition engine, said acquisition engine comprising a grammar layer and a comparison layer; and

generate structured data outputs from a cooperation of said grammar layer and said comparison layer, said grammar layer comprising a top-down parsing algorithm.

20. The system of Claim 19, wherein said at least one data example is one data example.

21. The system of Claim 19, wherein said example output is a context-free grammar and a pattern dictionary.

22. The system of Claim 21, wherein said context-free grammar further comprises regular grammar.

23. The system of Claim 19, further executable to separate said at least one data example and corresponding

coarse structure utilizing lexical analysis to form a first set of tokens and annotate said first set of tokens with characteristic features to produce said first plurality of annotated strings.

24. The system of Claim 23, wherein said predetermined input is a first HTML page and said first set of tokens is a first set of HTML phonemes.

25. The system of Claim 19, wherein said predetermined input is a PDF file or a semi-structured text file.

26. The system of Claim 19, wherein to generate said second plurality of annotated strings is further executable to preprocess said input from said semi-structured data sources to form said second plurality of annotated strings.

27. The system of Claim 26, wherein to preprocess is further executable to separate said input from said semi-structured data sources utilizing lexical analysis to form a second set of tokens and annotate said second set of tokens with characteristic features to produce said second plurality annotated strings.

28. The system of Claim 27, wherein said semi-structured data sources are one or more related HTML pages and said second set of tokens is a second set of HTML phonemes.

29. The system of Claim 19, wherein said acquisition of said at least one data example and said coarse structure from a predetermined input further comprises a user interface for identification of said at least one data example and said coarse structure.

30. The system of Claim 29, wherein said user interface is a Graphical User Interface (GUI) and said predetermined input is an HTML page.

31. The system of Claim 21, wherein to generate said pattern dictionary is further executable to assign a pattern consisting of a portion of said sequences of annotated strings to each of said at least one data example and assign additional patterns to distinguish said corresponding coarse structure from said predetermined input.

32. The system of Claim 31, wherein to generate said context-free grammar is further executable to generate terminals that are in one-to-one correspondence with said patterns in said pattern dictionary.

33. The system of Claim 19, wherein said cooperation of said grammar layer and said comparison layer is further executable to:

sequentially compare in said comparison layer said patterns in said pattern dictionary against the said second sequence of annotated strings to find a matching pattern in a portion of said second sequence of annotated strings;

compile a matching score representing a quality of a match between said patterns in said pattern dictionary and said matching pattern;

pass said matching score and said matching pattern to said grammar layer.

extend already found matching patterns with said matching pattern to form a sequence of matching patterns; and

execute a set of grammar rules defined by said context-free grammar on said sequence of matching patterns to locate a legal sequence of strings defined by said set of grammar rules and representing said structured data outputs.

34. The system of Claim 33 wherein to compile the matching score further executable to implement a weighted edit distance algorithm to calculate the matching score.

35. The system of Claim 34, wherein the weighted edit distance algorithm is a top down method with pruning.

36. The system of Claim 34, wherein the weighted edit distance algorithm is a dynamic programming method.

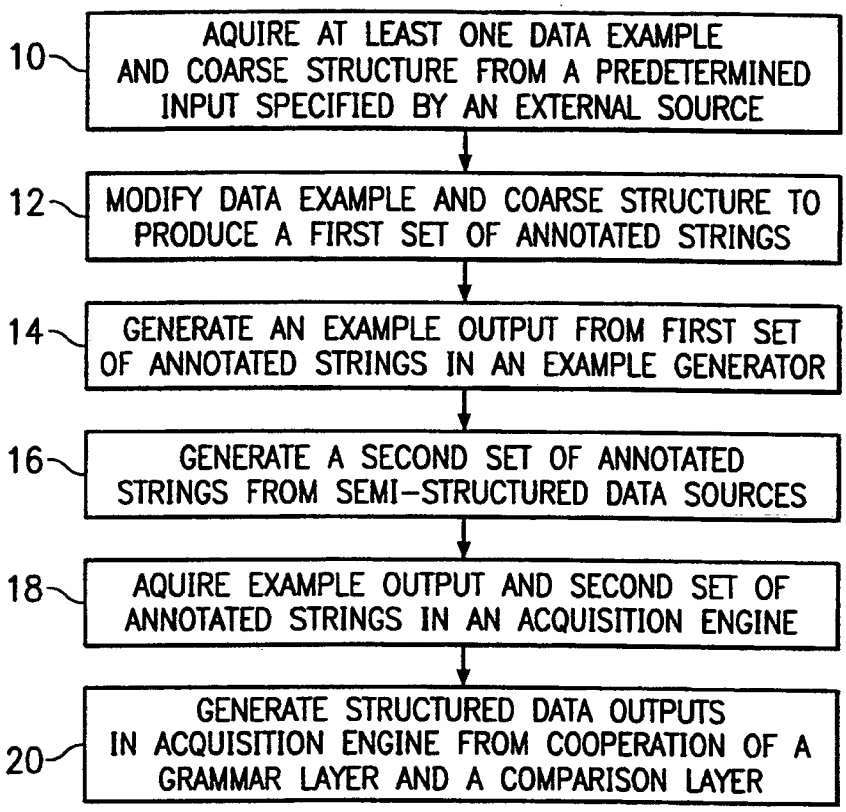

10
AQUIRE AT LEAST ONE DATA EXAMPLE AND COARSE STRUCTURE FROM A PREDETERMINED INPUT SPECIFIED BY AN EXTERNAL SOURCE
12
MODIFY DATA EXAMPLE AND COARSE STRUCTURE TO PRODUCE A FIRST SET OF ANNOTATED STRINGS
14
GENERATE AN EXAMPLE OUTPUT FROM FIRST SET OF ANNOTATED STRINGS IN AN EXAMPLE GENERATOR
16
GENERATE A SECOND SET OF ANNOTATED STRINGS FROM SEMI-STRUCTURED DATA SOURCES
18
AQUIRE EXAMPLE OUTPUT AND SECOND SET OF ANNOTATED STRINGS IN AN ACQUISITION ENGINE
20
GENERATE STRUCTURED DATA OUTPUTS IN ACQUISITION ENGINE FROM COOPERATION OF A GRAMMAR LAYER AND A COMPARISON LAYER

FIG. 1

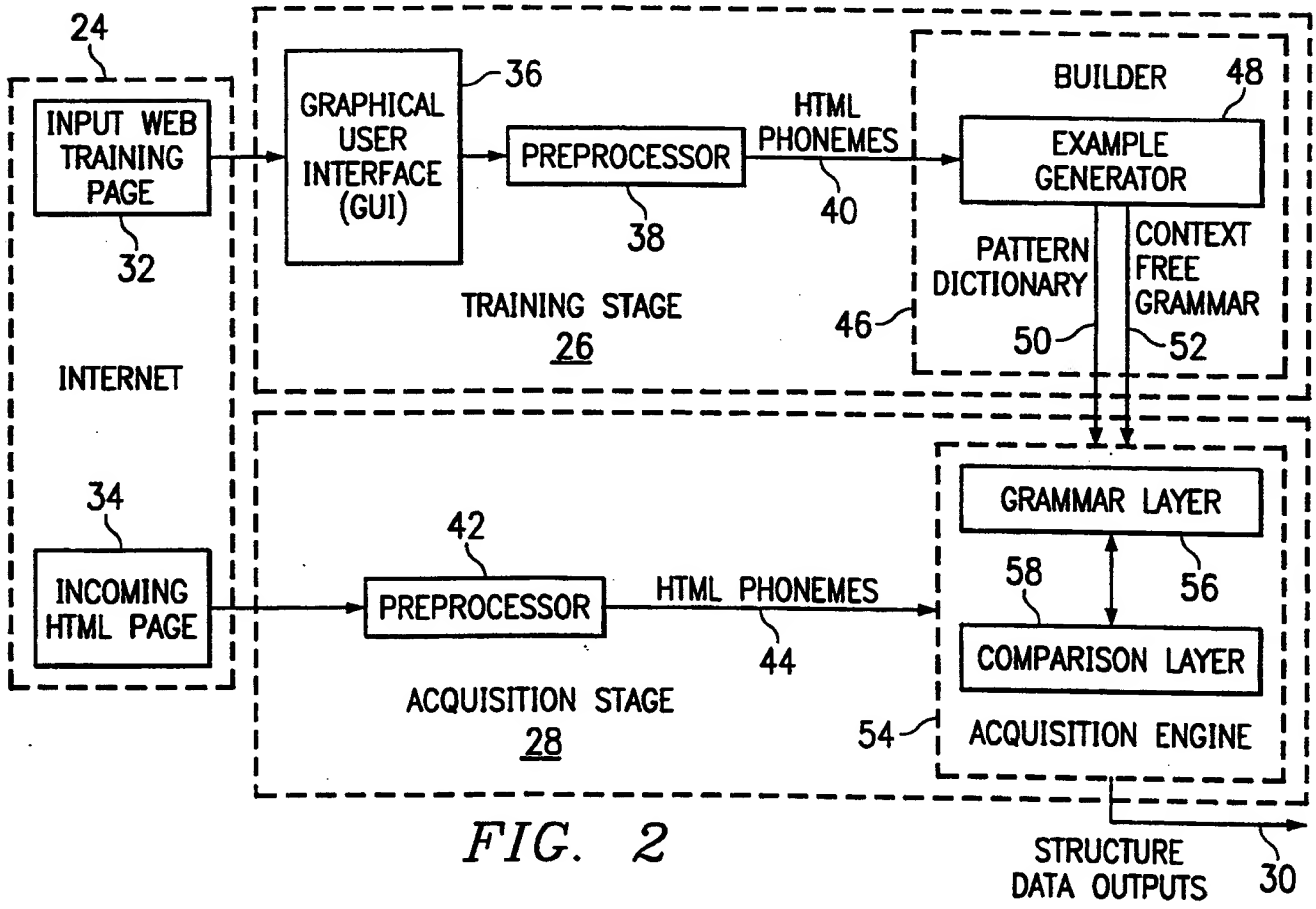

24
INPUT WEB TRAINING PAGE
32
GRAPHICAL USER INTERFACE (GUI)
36
PREPROCESSOR
38
HTML PHONEMES
40
BUILDER
48
EXAMPLE GENERATOR
PATTERN DICTIONARY
CONTEXT FREE GRAMMAR
46
50
52
TRAINING STAGE
26
INTERNET
34
INCOMING HTML PAGE
42
PREPROCESSOR
HTML PHONEMES
44
GRAMMAR LAYER
58
56
COMPARISON LAYER
ACQUISITION STAGE
28
54
ACQUISITION ENGINE
STRUCTURE DATA OUTPUTS
30

FIG. 2

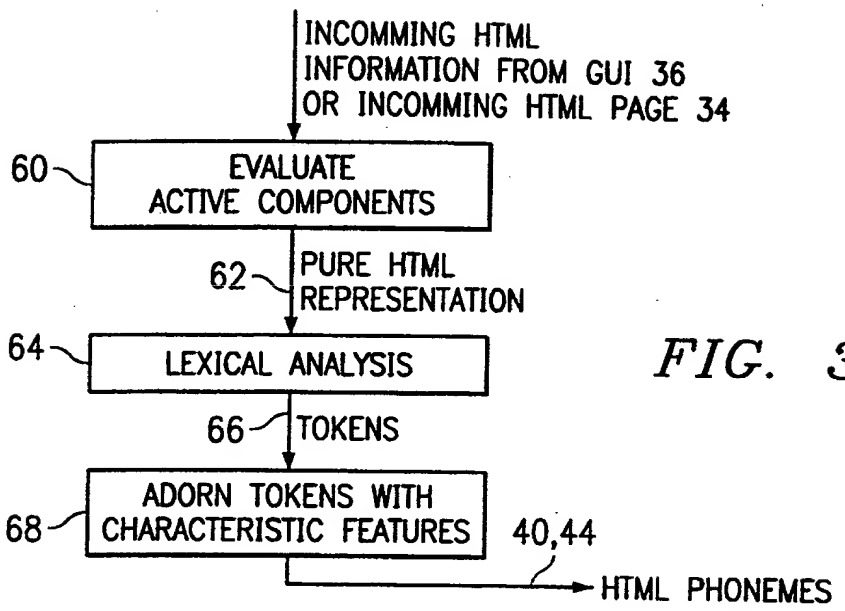
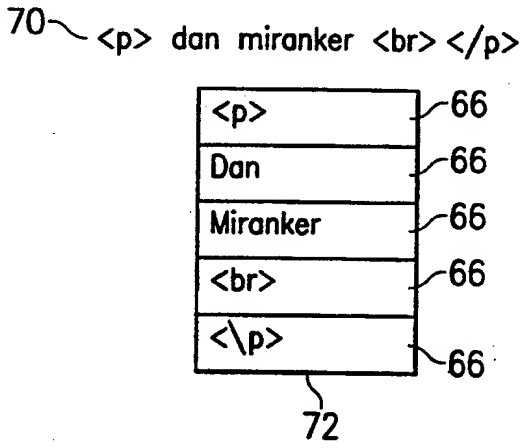


FIG. 3



74

	66 <p>	66 Dan	66 Miranker	66 	66 <\p>
CHARACTERISTIC FEATURE 1					
CHARACTERISTIC FEATURE 2					
⋮					
CHARACTERISTIC FEATURE N					

FIG. 4

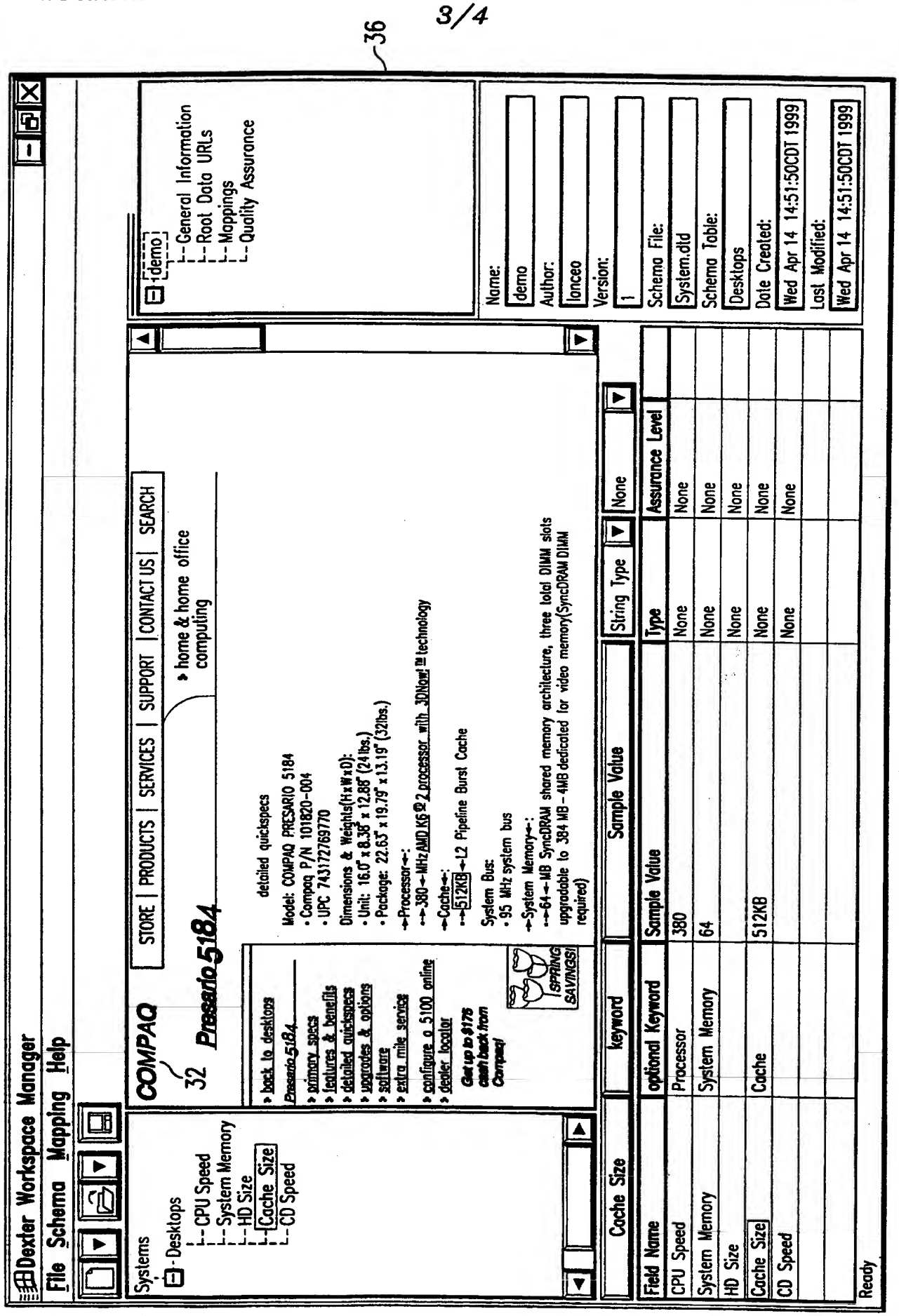

36
Dexter Workspace Manager
File Schema Mapping Help
Systems
Desktops
CPU Speed
System Memory
HD Size
Cache Size
CD Speed
COMPAQ
32
Presario5184
STORE | PRODUCTS | SERVICES | SUPPORT | CONTACT US | SEARCH
home & home office computing
back to desktops
Presario 5184
primary specs
features & benefits
detailed quickspecs
upgrades & options
software
extra mile service
configure a 5100 online
dealer locator
Get up to $175 cash back from Compaq!
SPRING SAVINGS!
detailed quickspecs
Model: COMPAQ PRESARIO 5184
• Compaq P/N 101820-004
• UPC 743172769770
Dimensions & Weights(HxWxD):
• Unit: 16.0" x 8.38" x 12.88" (24 lbs.)
• Package: 22.63" x 19.79" x 13.19" (32lbs.)
-Processor-:
--380--MHz AMD K6® 2 processor with 3DNow!™ technology
-Cache-:
--512KB--L2 Pipeline Burst Cache
System Bus:
• 95 MHz system bus
-System Memory-:
--64--MB SyncDRAM shared memory architecture, three total DIMM slots upgradable to 384 MB - 4MB dedicated for video memory(SyncDRAM DIMM required)
demo
General Information
Root Data URLs
Mappings
Quality Assurance
Name: demo
Author: lanceo
Version: 1
Schema File: System.dtd
Schema Table: Desktops
Date Created: Wed Apr 14 14:51:50CDT 1999
Last Modified: Wed Apr 14 14:51:50CDT 1999
Cache Size
keyword
Sample Value
String Type
None
Field Name | optional Keyword | Sample Value | Type | Assurance Level
CPU Speed | Processor | 380 | None | None
System Memory | System Memory | 64 | None | None
HD Size | | | None | None
Cache Size | Cache | 512KB | None | None
CD Speed | | | None | None
Ready

FIG. 5

50

PATTERN	PHONEMES WEIGHTS	CONTEXT FREE GRAMMAR TERMINAL
P0	p1w1 p2w2	E
P1		
Pm		

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/07792

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :Please See Extra Sheet.
US CL :Please See Extra Sheet.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/513, 102, 103

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	HUCK. G. Jedi:extracting and synthesizing information from the Web IEEE August 1998. pages 32-41.	1-36
Y	GRUSER. J. Wrapper generation for Web accessible data sources IEEE August 1998. pages 14-23.	1-15, 19-33
Y	WEIGEL. A. Lexical postprocessing by heuristic search and automatic determination of the edit costs IEEE August 1995. Vol 2. pages 857-860.	16-18, 34-36

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
19 MAY 2000	13 JUN 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MICHAEL RAZAVI Telephone No. (703) 305-3900

Form PCT/ISA/210 (second sheet) (July 1998) ★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/07792

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (7):

G06F 15/00, 17/00, 17/21, 17/24, 7/00

A. CLASSIFICATION OF SUBJECT MATTER:
US CL :

707/513, 102, 103

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

WEST IEEE, ACM
search terms: HTML, data extraction, OLAP, wrapper, JEDI, data mining